Categorization and Modelling of Quality in Context Information

M.A. Razzaque, Simon Dobson and Paddy Nixon
Systems Research Group
School of Computer Science and Informatics
University College, Dublin IE
abdur.razzaque@ucd.ie, simon.dobson@ucd.ie, paddy.nixon@ucd.ie

Abstract: *Pervasive Computing* environments are dynamic and heterogeneous. They are required to be self-managing and autonomic, demanding minimal user's guidance. In pervasive computing, context-aware adaptation is a key concept to meet the varying requirements of different clients. In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. It is clear that some form of *context categorization* will be required given the wide range of heterogeneous context information. Categorizations can be made from different viewpoints such as conceptual viewpoint, measurement viewpoint, temporal characteristics viewpoint and so on. To facilitate the programming of context-aware applications, *modelling of contextual information* is highly necessary. Most of the existing models fail both to represent *dependency relations* between the diverse context information, and to utilize these dependency relations. A number of them support narrow classes of context and applied to limited types of application, and most do not consider the issue of *Quality of Contextual Information* (QoCI). Along with a detailed context categorization, this paper will analyse existing context models and discuss their handling of dependency issues. It uses this analysis to derive a methodology for quality context information modelling in context aware computing.

1. Introduction

Pervasive Computing envisages a world with users interacting naturally with device-rich environments to perform a variety of tasks [Streitz and Nixon, 2005]. These environments are dynamic and heterogeneous. They are required to be self-managing and autonomic; demanding minimal user's guidance. In this heterogeneous environment of *Pervasive Computing*, context-aware [Coutaz et al., 2005] adaptation is a key concept to meet the varying requirements of different clients. In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. It is clear that some form of *context categorization* will be required given the wide range of heterogeneous context information. Two important categorizations viewpoints are:

- *Conceptual viewpoint* – who, where, what occurs, when, what can be used, what can be obtained etc.
- *Measurement viewpoint* – what is the room temperature or network bandwidth or network latency etc?

To facilitate the programming of context-aware applications an infrastructure is necessary to gather, manage and disseminate context information to applications. And this infrastructure ultimately requires the *modelling of contextual information.* There are number of existing context descriptions based on one of the following methods:

- Set theory
- Directed Graph
- First-order Logic
- Preference and user Profiles

Most of these models fail to both represent *dependency relations* between the diverse context information and to utilize these dependency relations. A number of these support narrow classes of context and applied to limited types of application. Furthermore most of them do not consider the issue of *Quality of Contextual Information* (QoCI). This will be a critical issue for the next generation pervasive computing; primarily because the quality of a given piece of contextual information will dramatically effect the decisions made by the autonomous application. Along with a detail context categorization this paper will analyse existing context models. Dependency relations, one of the missing issues in most of the existing context model are discussed. Further it presents a methodology for quality context information modelling in context aware computing.

The organization of the paper is as follows. Section 2 defines what we mean by context and context awareness. Context categorization and analysis of context models are presented in section 3 and section 4 respectively. Section 5 briefly describes the dependency relations in context information. A methodology of quality context information is presented in section 6, while section 7 concludes with some future directions.

2. What is context and context awareness?

It is quite unlikely that a single definition of context will be accepted by all researchers. From time to time, from application to application this definition varies. Historically [Winograd, 2001], "Context" has been adapted from linguistics, referring to the meaning that must be inferred from the adjacent text. In respect to computing world definitions of context varies with *computing environment* (available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing) *user environment* (location, collection of nearby people, and social situation) and *physical environment* (lighting, noise level etc). According to [Dey et al., 2000a] context is "*any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects*." Although this definition encompasses the definitions given by previous authors, it is sometimes too broad. [Winograd, 2001] has given a more specific and role based definition. According to him context "*is an operational term: something is context because of the way it is used in interpretation, not due to its inherent properties.*" Most recently [Coutaz et al., 2005] defined context "*is not simply the state of a predefined environment with a fixed set of interaction resources. It's part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources.*"

Context awareness is a term from computer science, which is used for devices that have information about the circumstances under which they operate and can react accordingly. Context-aware computing involves application development that allows for collection of context and dynamic program behavior dictated by knowledge of this environment. Context-awareness is not unique to ubiquitous computing. For example, explicit user models used to predict the level of user expertise or mechanisms to provide context-sensitive help are good examples used in many desktop systems. With increased user mobility and increased sensing and signal processing capabilities, there is a wider variety of context available to tailor program behavior. Through context-awareness rapid personalization of computing services will be possible.

Today's computer systems are unaware of the user's context. They do not discern what the user is doing, where is the user, who is nearby and other information related to the user's environment. They just take the explicit input from the user, process it, and then output the result. Deemed as computing for the next generation, pervasive computing will greatly change the way today's computers behave. The basic idea is to instrument the physical world around us with various kinds of sensors, actuators, and tiny computers. The huge amount of information can then be collected and processed by computer systems, enabling computer systems to deduce the user's situation and act correspondingly with user's intervention [Nixon et al, 2002]. Active Badge System, Call Forwarding, Teleporting, PracTab system, Conference Assistant, Office Assistant, Classroom 2000, CyberDesk, etc are examples of present context aware Systems/Applications.

Category	Semantics	Examples
User context	Who?	*User's Profile:* identifications, relation with others, to do lists, etc
Physical context	Where?	*The Physical Environment:* humidity, temperature, noise level, etc
Network context	Where?	*Network Environment:* connectivity, bandwidth, protocol, etc
Activity context	What occurs, when?	*What occurs, at what time:* enter, go out, etc
Device context	What can be used?	*The Profile and activities of Devices:* identifications, location, battery lifetime, etc
Service context	What can be obtained?	*The information on functions which system can provide:* file format, display, etc

Table 1: Conceptual Categorization

3. Context Categorization

Context categorization will be required for the wide range of heterogeneous context information in next generation context aware computing. Context categorization helps application designer and developer to uncover the possible context and simplify the context manipulation. Classification context information can be helpful in providing quality context information. For example, conflicts can be resolved by favoring the classes of context that are most reliable (static followed by profiled) over those that are more often subject to error (sensed and derived).

Two possible broad categorizations viewpoints are:

- *Conceptual viewpoint* – who, where, what occurs, when, what can be used, what can be obtained etc.
- *Measurement viewpoint* – what is the room temperature or network bandwidth or network latency etc?

But most of the researchers did the categorization from conceptual viewpoint and some of them are following:

- [Gwizdka, 2000]
    - Internal Context: the state of the user
    - External context: the state of the environment
- [Petrelli et al., 2000]
    - Material Context: the location, device and available infrastructure
    - Social Context: social aspects and personal traits
- [Dey et al., 2000a]
    - Primary Context: location, time and activity
- [Schilit et al., 1994]
    - Primary Context: user environment, physical environment, computing environment

Although aforementioned categorizations are helpful but sometimes context information can't be clearly delimited and they are incomplete. Considering these issues this paper is aimed to provide a more comprehensive categorization from conceptual viewpoint as well as from measurement viewpoint.

Conceptual categorization:

The conceptual categorization of context (table 1) provides a description of the contextual space in terms of the actors, the actions and the relationships between them.

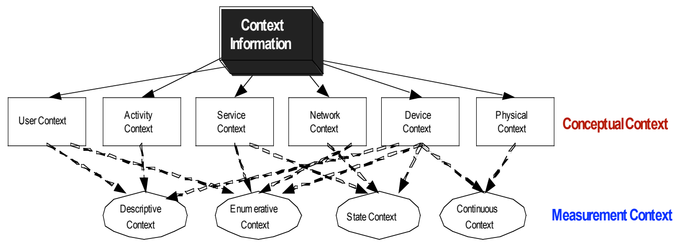

Figure 1: Hierarchical Categorization of Context information

Measurement Categorization:

- Continuous Context
- Enumerative Context
- State Context
- Descriptive Context

Continuous Context

In this category the value of context changes continuously. Continuous context component (ξ), is function of

-- current value of the context component,
-- lowest threshold value
-- highest threshold value
-- compare value
-- the metric of the value

and it uses function formula for the calculation.

Enumerative Context

Here the values of context are a set of discrete values and defined in a list or set. They are based on set operations. Like, enumerative context componentδ, **val** $(\delta)\in\Delta$, $\Delta = \{\delta_1 \ldots \delta_i \ldots . \delta_n\}$

State Context

This category consists of two opposite values and they toggle between them. Like, state context component η **val** $(\eta)\in H$, $H=\{0,1\}$ and this is calculated in predicate calculus.

Descriptive Context

This is based on the description statement of the context and for this purpose it uses predicate calculus. For example;

location (CellPhone, loc_A)
location(laptop, loc_B)
location (obj1, loc1) ^ location (obj2, loc2) ^ (loc1^ loc) ^ (loc2 ^ loc) =>near(obj1,obj2)

Another context categorization could be done in terms of temporal properties of context information:

- **Static context:** Static context information describes those aspects of a pervasive system that are invariant, such as a person date of birth, social security number etc.
- **Dynamic context:** Pervasive systems are typically characterized by frequent changes; the majority of information is dynamic. The persistence of dynamic context information can be highly variable; for example, relationships between colleagues typically last for months or years, while a person's location and activity often change from one minute to the next.

Conceptual and measurement viewpoints contexts could be again classified as static or dynamic contexts. Above categorizations are not exhaustive for future's pervasive computing where context information will exhibit more diverse characteristics but these could be very helpful for application designer and developer in pervasive computing to manipulate context information efficiently.

4. Context Modeling

To facilitate the programming of context-aware applications an infrastructure is necessary to gather, manage and disseminate context information to applications. And this infrastructure ultimately requires the *modeling of contextual information.* Context modeling is highly important to capture:

- user requirements/profile, application requirements, device capabilities
- relationship between context

Context information is gathered, stored, and interpreted at different parts of the system. A representation of the context information should be applicable throughout the whole process of gathering, transferring, storing, and interpreting of context information. Most of the existing context models are based on one of the following methods:

- Set theory
- Directed Graph
- First-order Logic
- Preferences and user's Profiles (CC/PP and CSCP)

Set theory

- [Schmidt et al., 1999] used set theory for the context presentation. The context *T* is described by a set of two-dimensional vectors. Each vector *h* consists of a symbolic value *v* describing the situations and a number *p* indicating the certainty that the user (or the device) is currently in this situation.
- [Yau et al, 2001] also used set theory for the context and a *context-tuple* is defined as a tuple <ai, aj, ak, . .. an, tm> of size *n,* where *n* is the number of unique contextual-data sources present in the device. Each variable *ai* in the tuple represents a value, which is valid for the corresponding type of context. The variable *tm* represents the time of the tuple creation time.

Set theory describe context schematically and dependency relations are not embodied.

Directed Graph

[Henricksen et al., 2002] proposed an object-based context modeling in which context information is structured around a set of entities, each describing a physical or conceptual object such as person or communication channel. It uses the form of a directed graph for the diagrammatic representation of context, in which entity and attribute types form the nodes, and associations are modeled as arcs connecting these nodes. This is a comprehensive model which includes QoCI and dependency relations but fails to represent the dependency relation accurately.

First-order Logic

[Ranganathan et al., 2002] proposed a context model named *ConChat* and it is based on first-order predicate calculus and Boolean algebra. It covers the wide variety of available contexts and supports various operations, such as conjunction and disjunction of contexts and quantifiers on contexts. It allows the creation of complex first-order expressions involving context, so it is possible to write various rules, prove theorems, and evaluate queries. This modeling is consists of the four elements in the following ways:

- Context (<ContextType>, <Subject>, <Relater>, <Object>)

ContextType: the type of context,
Subject: person, place, or thing, with which the context is concerned,
Object: a value associated with the subject,
Relater: comparison operator, verb, or preposition
Examples:
context(people, Room 22,>=,3)
context(application, PowerPoint, Is, Running)
context(RoomActivity, 22, Is, Presentation)

This is a well defined modeling to specific field like electronic chat but in this model relation between continuous data cannot be described easily and even it is not dealing with QoCI.

Preferences and user Profiles

Composite Capability/Preference Profiles (CC/PP) [Klyne et al., 2001] is the W3C's proposal for a profile representation language and it is a framework based on the Resource Description Framework (RDF). CC/PP is intended to express both device capabilities and user preferences. Its specification defines a basic structure for **profiles.** A profile is basically constructed as a strict two-level- hierarchy: each profile having a number of **components**, and each component having a number of **attributes** (shown in figure 2). The particular components and attributes are not defined by the CC/PP specification. The definition of a specific vocabulary is up to other standardization bodies. Although CC/PP able to fulfill all the requirements except structural property of profile representation mentioned [Held et al., 2002] but vocabulary is not rich enough; it needs to be extended. Most importantly it can't represent the complex relationships and constraints. Even Component/Attribute model becomes clumsy if there are many layers.

Comprehensive Structured Context Profiles

Comprehensive Structured Context Profiles (CSCP) [Held et al., 2002] is based on the Resource Description Framework (RDF) and overcomes the shortcomings of the Composite Capability/Preference Profiles language (CC/PP) regarding structuring. Furthermore it extends the mechanisms to express user preferences. It can't represent the complex relationships and constraints. Component/Attribute model becomes clumsy if there are many layers.

From the above study it is quite clear that existing context models are suffering at certain extent which makes them not very suitable as a context model for future pervasive systems. Future's full fledged pervasive systems will require much more sophisticated context models in order to support seamless adaptation to changes in the computational environment. The context models will need to specify a range of characteristics/quality of context information including temporal characteristics (freshness and histories) accuracy resolution (granularity) confidence in correctness of context information, as well various types of dependencies among the different context information.

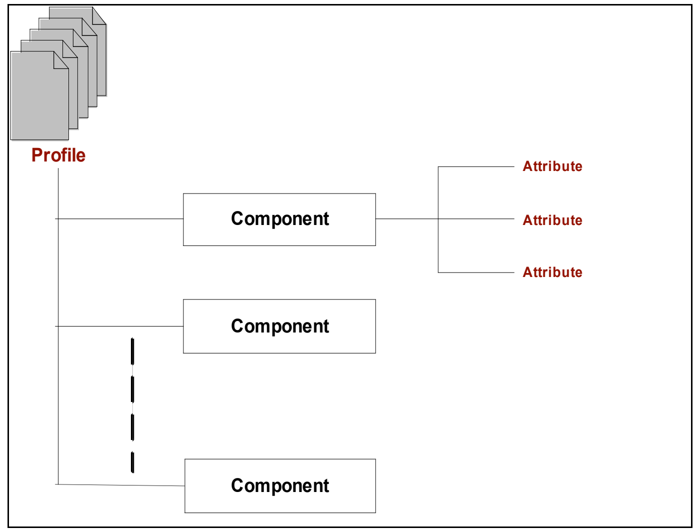

Figure 2: CC/PP

5. Dependency relations

Future pervasive and context aware systems will need to deal with heterogeneous services and contexts. It is very likely that these context information will be some how interrelated and dependent. According to [Henricksen et al., 2002], "*A dependency is a special type of relationship, common amongst context information, which exists not between entities and attributes, as in the case of associations, but between associations themselves*." Here associations are the unidirectional relationships between the entity and its attributes and a dependency shows the reliance of one association upon another. [Efstratiou et al., 2001] showed the importance of capturing dependencies in context aware applications. Without knowledge of such dependencies, inappropriate decisions can be made by context-aware applications that lead to instability and unwanted results. Moreover, knowledge of dependencies is important from a context management perspective, as it can assist in the detection of context information that has become out-of-date**.** Dependency relations will be critical in diverse context information and it can't be ignored most of the cases. Above analysis on the number of existing context models shows that they don't include these dependency relations and suffer for this issue. Hence future context models should include these dependency relations more comprehensively.

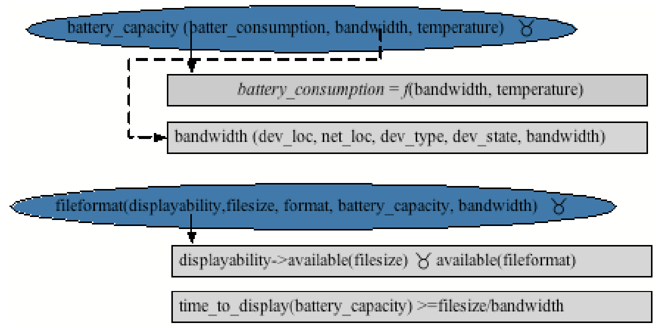

Figure 3: Dependency Description

Constraint Logic Programming Language [Marriott 1998] is a language, which allows the programmer simply to state relationships between objects and this, could be used for the description dependency relation. Constraint languages provide powerful, high-level descriptions for rule-based systems modelling which can operate on different types of (primary and derived) data. Consider, for example, displaying information in a smart phone like Nokia 6630. Figure 3 shows a sample scenario of the dependency description related to display information in a smart phone where two main concerns are battery power and file format.

6. Quality of Context Information (QoCI)

In context aware systems, errors in context information may arise as a result of errors in *gathering (sensing), interpretation* and *presentation* level. As context information is relied upon by applications to make decisions on the user's behalf, it is indispensable that applications have some means by which to judge the reliability of the information. *Quality of Contextual Information* or *data* is a judgment parameter or criteria for the contextual information or data. Most of the existing context models do not consider the issue of *Quality of Contextual Information* (QoCI). This will be a critical issue for the next generation pervasive computing; primarily because the quality of a given piece of contextual information will dramatically effect the decisions made by the autonomous application. Poor information or data quality can have severe impact on the overall effectiveness of the context aware system. Therefore inclusion of QoCI in the future context model is highly necessary.

Next generation pervasive and context aware systems will need to deal with heterogonous applications which will require diverse context information. Moreover these assorted applications will require various *Quality of Service (QoS)*.To provide these QoS we need various *QoCI* to be incorporated in the context model.

Before analyzing or managing information or data quality, one must understand what information or data quality means. Information quality management requires understanding which dimensions of information quality are important to the user or application. According to [Wang et al., 1993] we can define *QoCI* in terms of information quality parameters and information quality indicators as below:

- ***An information quality parameter*** is a qualitative or subjective dimension by which a user evaluates context information quality. *Source credibility* and *timeliness* are examples.
- ***An information quality indicator*** is a context information dimension that provides objective information about the context. *Source, creation time, and collection method are examples.*
- ***An information quality attribute*** is a collective term including both quality parameters and quality indicators.
- ***An information quality indicator value*** is a measured characteristic of the gathered and stored data. The information quality indicator source may have an indicator value like from a sensor or user.
- ***An information quality parameter value*** is the value determined for a quality parameter (directly or indirectly) based on underlying quality indicator values. Application-defined

functions may be used to map quality indicator values to quality parameter values. For example, because the *source is user himself for his date birth information,* so *credibility* is high.

- ***Information quality requirements*** specify the indicators required to be tagged, or otherwise documented for the information related to an application or group of applications. If a context model includes this then it is possible to make the context aware system more efficient and effective.

Necessity of the diverse quality of context information has been broadly recognized in number of research works, yet none of the existing work addresses the problem in an adequate or general way. [Dey et al., 2000b] suggests that ambiguity in information can be resolved by a mediation process involving the user. But in case of potentially large quantities of context information involved in pervasive computing environments and the rapid rate at which context can change, this approach places an unreasonable burden on the user. [Ebling et al., 2001] describe a context service that allows context information to be associated with quality metrics, such as freshness and confidence, but their model of context is incomplete and lacks formality. [Castro et al., 2001] defined notion of quality based on measures of accuracy and confidence, but their work limited to location information. Schmidt et al. associates each of their context values with a certainty measure that captures the likelihood that the value accurately reflects reality [Schmidt et al, 1999] .They are concerned only with sensed context information, and moreover take a rather narrow view of context quality. Gray and Salber include information quality as a type of meta-information in their context model, and describe six quality attributes: coverage, resolution, accuracy, repeatability, frequency and timeliness [Gray et al., 2001]. Finally [Henricksen 2002] included QoCI in their directed graph based context model but this could be limited to this sort of modelling. Most of their quality models are not formally defined, as they are intended to support requirements analysis and the exploration of design issues, rather than to support the development of a context model that can be populated with data and queried by applications.

Considering the above limitations in quality modelling our effort is to provide a generic approach of quality context information modelling based on [Wang et al., 1993]. Figure 4 shows the step by step methodology for quality contextual information modelling where initial input is *user's and corresponding application's* requirements and the final outcome of the modelling is the *quality schema.* Each step includes the *input, process* and *output*. Table 2 provides a brief description of each step:

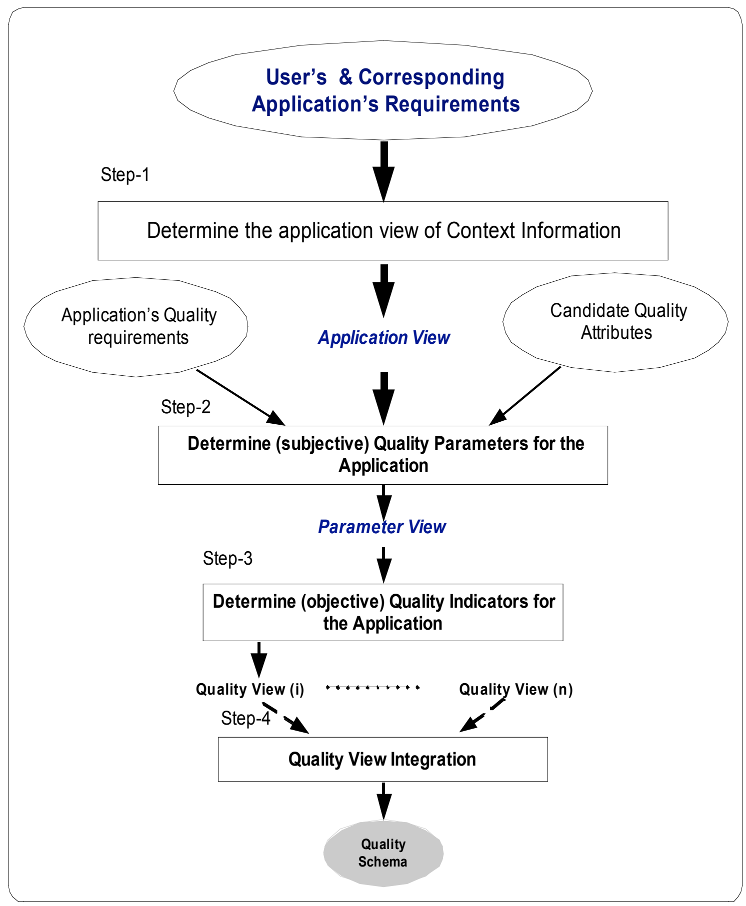

Figure 4: The process of quality contextual information modelling

Step No.	Input	Output	Process
Step-1	User's and Corresponding Application's requirements	Application view	It embodies traditional context information modelling and objective is to extract and document application requirements of context information.
Step-2	Application view, application quality requirement, candidate quality attributes	Parameter view	It determines the quality parameters (like timeliness, reliability etc) to support information quality requirements.
Step-3	Parameter view(application view included quality parameters)	Quality view	It converts the subjective quality parameters into measurable characteristics or quality indicators (like timeliness to date, etc)
Step-4	Quality view/views	Quality schema	This involves the integration of quality indicators.

Table 2: Brief description of the methodology for quality contextual information modelling

7. Conclusion

Next generation context aware systems have to deal with diverse context information. Categorization of this context information will be helpful for the context aware application designers and developers. To address this issue, this paper deals with categorizations and quality modeling in context information. Categorizations can me made from different viewpoints such as conceptual viewpoint, measurement viewpoint, temporal characteristics viewpoint. To facilitate the programming of context-aware applications, *modelling of contextual information* is highly necessary. An analysis of the number of existing models shows most of these models fail to both represent *dependency relations* between the diverse context information and to utilize these dependency relations. A number of these support narrow classes of context and applied to limited types of application. Moreover most of them do not consider the issue of *Quality of Contextual Information* (QoCI). A methodology for quality contextual information modelling in context aware computing is presented. The methodology is briefly described. Detail of quality modeling in contextual information with details of different application oriented quality dimensions can be extended in future work.

References

[Castro et al., 2001] Castro,P.,Chiu, P.,Kremenek, T.,Muntz,R. "A probabilistic room location service for wireless networked environments" UbiComp 2001 Conference, Atlanta (2001)

[Coutaz et al., 2005] Joëlle Coutaz, James Crowley, Simon Dobson, and David Garlan. "Context is key." Communications of the ACM, 48(3), March 2005

[Dey et al, 2000a] A.K.Dey, G. D.Abowd. "Towards a Better Understanding of Context and Context-Awareness."CHI2000 Workshop, 2000.

[Dey et al, 2000b] Dey,A., Manko.,J., Abowd,G. "Distributed mediation of imperfectly sensed context in aware environments."Technical Report GIT-GVU-00-14,Georgia Institute of Technology (2000)

[Ebling et al., 2001] Ebling,M.,Hunt,G.D.H.,Lei,H. "Issues for context services for pervasive computing." Middleware 2001 Workshop on Middleware for Mobile Computing,Heidelberg (2001)

[Efstratiou et al., 2001] Efstratiou, C.,Cheverst,K.,Davies,N.,Friday,A. "An architecture for the effective support of adaptive context aware applications.In:Mobile Data Management (MDM)"Hong Kong,China,Springer (2001)15-26.

[Gray et al., 2001] Gray,P.,Salber,D. "Modelling and using sensed context in the design of interactive applications." In 8th IFIP Conference on Engineering for Human-Computer Interaction, Toronto (2001)

[Gwizdka, 2000] J.Gwizdka. "What's in the Context."CHI2000 Workshop.

[Held et al.,2002] Held, A., Buchholz, S., Schill, A. "Modeling of Context Information for Pervasive Computing Applications" Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI2002), Orlando, FL, Jul 2002

[Henricksen et al., 2002] K.Henricksen, J.Indulska, A.Rakotonirainy. "Modeling Context Information in Pervasive Computing Systems."Proceedings Pervasive 2002 -Zurich August 2002.

[Indulska et al., 2003] Jadwiga Indulska, Ricky Robinson, Andry Rakotonirainy, Karen Henricksen. "Experiences in Using CC/PP in Context-Aware Systems."Proceeding of Mobile Data Management. Jan. 2003.

[Klyne et al., 2001]G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies", *W3C Working Draft*, Mar 15, 2001.

[Marriott and Stuckey, 1998] Kim Marriott, Peter J. Stuckey. "Programming with Constraints: An Introduction."MIT Press. 1998.

[Streitz and Nixon, 2005] Norbert Streitz and Paddy Nixon, "The Disappearing Computer", Special issue of Communications of the ACM, 48(3), March 2005.

[Nixon et al., 2002] Paddy Nixon, Feng Wang, Sotirios Terzis and Simon Dobson. Engineering context-aware systems. In Proceedings of the International Workshop on Engineering Context-Aware Object-Oriented Systems and Environments. 2002.

[Petrelli et al, 2000] D. Petrelli, E. Not, C. Strapparava, O. Stock, M. Zancanaro. "Modeling Context is Like Taking Pictures."CHI2000 Workshop, 2000.

[Ranganathan et al., 2002] Anand Ranganathan, Roy H. Campbell,Arathi Ravi,Anupama Mahajan. "ConChat: A Context-Aware Chat Program."IEEE Pervasive Computing. Vol.1,Iss.3, July-Sept. 2002. p51 –57.

[Schilit et al., 1994] B. Schilit, N. Adams, R. Want. "Context-aware computing applications."Procsof IEEE workshop on Mobile Computing Systems and Applications. 1994. p85-90.

[Schmidt et al, 1999] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, et.al. "Advanced Interaction in Context."1stInternational Symposium on Handheld and Ubiquitous Computing (HUC'99).

[Wang et al., 1993] Wang, R.Y.; Kon, H.B.; Madnick, "Data Quality Requirements Analysis and Modeling" Data Engineering, 1993. Proceedings. Ninth International Conference on 19-23 April 1993 Page(s):670 - 677

[Winograd, 2001] T. Winograd, "Architecture for Context", Human Computer Interaction , Vol. 16,pp401-419,2001 .

[Yau et al, 2001] Stephen S. Yau, Fariaz Karim. "Context-Sensitive Middleware for Real-time Software in Ubiquitous Computing Environments."Proceedings. 4th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC2001). May 2001. p163 –170.

[Zaslavsky, 2002] Arkady Zaslavsky."Adaptability and Interfaces: Key to Efficient Pervasive Computing."NSF workshop series on Context-Aware Mobile Database Management.Jan.2002.